

REMARKS

The Examiner is thanked for the performance of a thorough search. By this response, Claims 30–32 are added. No claims are amended or canceled. Hence, Claims 1–32 are pending in this application. The added claims do not add any new matter to this application and are supported by the Specification.

All issues raised in the Office Action mailed December 15, 2008 are addressed hereinafter.

I. INTERVIEW SUMMARY

Applicants thank the Examiner for the telephone interview conducted on March 11, 2009. Examiner McLean represented the USPTO. Applicants were represented by Karl T. Rees and Marcel Bingham. The parties discussed Claim 1 in view of the *Schwier* reference. In particular, Applicants pointed out that none of the input or output documents in *Schwier* are in a “merge format.” In response, the Examiner suggested that Applicants submit the arguments in a formal response to give the Examiner time to better consider *Schwier* and other references. No agreement on the allowability of Claim 1 was reached.

II. CLAIM REJECTIONS BASED ON 35 U.S.C. § 102

Claims 1–29 were rejected under 35 U.S.C. § 102(b) as allegedly unpatentable over U.S. Patent No. 7,202,972 (hereinafter “*Schwier*”). Applicants traverse the rejection. Reconsideration is respectfully requested.

CLAIM 1

Claim 1, recites a method useful for, among other purposes, a merge utility executing at a computer to generate print-ready documents that are in a “merge format” (e.g. PCL, Postscript, etc.). To this end, the merge utility receives two documents—a “first merge document” that is already in a merge format and a “second document” that is in an original format (e.g. a Word document). In response to receiving these two documents, the merge utility causes the second

document to be converted to the merge format (e.g. by converting the document itself, or by requesting that another application convert the document). The merge utility then generates a “composite merge document” by combining the first document and the converted second document.

Specifically, the method of Claim 1 features, among other steps:

- receiving . . . a first merge document that is in a merge format;
- converting a second document from an original format to the merge format to create a second merge document;
- . . .
- . . . merging the first merge document and the second merge document to generate a composite merge document; and
- after generating the composite merge document, delivering said composite merge document to an output device;
- . . .
- wherein the original format is a format that is not supported by the output device, and therefore needs to be converted to another format that is supported by the output device in order to be properly interpreted by the output device; and
- wherein the merge format is a format that is supported by the output device, and therefore does not need to be converted to another format that is supported by the output device in order to be properly interpreted by the output device.

By contrast, to the extent relied upon by the Office Action, *Schwier* describes an application 10 which accepts as input variable data 11 and static data 12. *Schwier* at FIG. 1. The application 10 then outputs a single document, V+S 13. Neither the inputs nor the outputs to application 10 are in a merge format.

The Office Action is therefore in error for at least the following reasons:

(1) *Schwier*’s application does not output a document in the merge format

As mentioned above, Claim 1 recites that a merge utility “merg[es] the first merge document and the second merge document to generate a composite merge document.” The composite merge document is recited to be in a “merge format,” which is “a format that is supported by the output device, and therefore does not need to be converted to another format that is supported by the output device in order to be properly interpreted by the output device.”

The Office Action alleges that *Schwier*'s V+S 13 is a "composite merge document" generated by a merge utility within the meaning of Claim 1. The Office Action is mistaken.

Schwier's V+S 13 is very clearly described as being in EMF format. *See, e.g.*, FIG. 2; col. 6, lines 7–14. One skilled in the art would not understand EMF to be a merge format because EMF is not "supported by the output device"—e.g. the printer. This point is abundantly clear in *Schwier*, which describes how the EMF document is subsequently sent to a PCL converter 18 (or EMF->PCL Convertor 58) so as to be converted to PCL before printing. *See, e.g.*, FIG. 2; col. 7, lines 7–17; col. 8, lines 13–15; col. 9, lines 64–65.

Therefore, *Schwier*'s V+S 13 is not a "composite merge document" within the meaning of Claim 1. Nor is *Schwier*'s application 10 a "merge utility" within the meaning of Claim 1 because it fails to generate a composite merge document.

(2) *Schwier*'s application does not input any document that is in a merge format

Even if *Schwier*'s application 10 were to generate a "composite merge document" within the meaning of Claim 1, application 10 would still not be a merge utility within the meaning of Claim 1 because application 10 does not input any document that is in a merge format. As quoted above, Claim 1 recites a step of "receiving . . . a first merge document that is in a merge format." The Office Action appears to allege that one of Variable Data 11 or Static Data 12 corresponds to the "first merge document." The Office Action is mistaken, for at least the reason that neither Variable Data 11 nor Static Data 12 is in a "merge format."

Schwier teaches that Variable Data 11 is stored in "a separate datafile [such as] a Word document, data bank, [or] an Excel document." *Schwier* at col. 5, lines 35–38. None of the example data files that *Schwier* describes as being used to store Variable Data 11 is in a "merge format." Nor are Applicants aware of any data storage technique that relies on storing variable data in such "merge formats" as PCL or Postscript. Thus, Variable Data 11 is not in a merge format within the meaning of Claim 1.

Schwier teaches that Static Data 12 is comprised of "static data areas" in a "master document" produced in a Winword program 10. *Schwier* at col. 5, lines 32–35. Applicants are unaware of any Winword program capable of "producing" a document that is in a merge format. Thus, Static Data 12 is not in a merge format within the meaning of Claim 1.

Nor could *Schwier*'s application 10 receive documents in a merge format, as required of the merge utility recited in Claim 1. Application 10 is specifically described as being a "Winword program." *Schwier* at col. 5, lines 34. Applicants are unaware of any Winword program capable of "receiving" a document in a merge format. Certainly the example Winword program of Word 97 would have been unable to receive and interpret such formats as Postscript and PCL.

Nor would it be obvious to modify *Schwier*'s application 10 to receive a document in a merge format. Rather, it would be counterproductive for application 10 to receive data that was already in a print-ready format, convert it to a non-print format (EMF), and then convert it back to a print-ready format again. If *Schwier* had intended for either of Variable Data 11 or Static Data 12 to be stored in a "merge format," *Schwier* would have devised a very different technique for optimizing his printing process, so as to avoid such unnecessary conversion of data.

Therefore, *Schwier* does not teach or suggest that application 10 "receive[es] a first merge document," as recited in Claim 1. Application 10 is therefore not a "merge utility."

(3) *Schwier*'s application does not convert any document to a merge format prior to combining it with another document

Finally, Claim 1 recites a step of "converting a second document from an original format to the merge format to create a second merge document." This step occurs before generating a composite merge document, as the composite merge document is generated by merging the first merge document with the newly created second merge document.

Even if *Schwier* were to teach that application 10 "receive[es] a first merge document," *Schwier*'s application 10 would still fail to teach or suggest the "merge utility" of Claim 1 because *Schwier*'s application 10 does not "convert[] a second document from an original format to the merge format to create a second merge document" prior to generating the alleged "composite merge document" V+S 13. The Office Action alleges that *Schwier* teaches this step because an Excel document may be converted to PCL. The Office Action is mistaken.

While one skilled in the art may have known that an Excel document can be converted to PCL, *Schwier* does not teach that application 10 converts an Excel document to PCL prior to combining the converted Excel document with a "first merge document." Rather, at best, *Schwier* teaches that data may be pulled from an Excel document into variable data areas of a

master document, and that the master document may then be converted to an EMF document. *Schwier* at col. 5, lines 32–38; col. 6, lines 8–14. This process is very different than that recited in Claim 1.

A proper anticipation rejection requires a single prior art reference to disclose each and every feature of a claim, *arranged as in the claim*. *E.g.* *Net Moneyin, Inc. v. Verisign, Inc., et al.* No. 2007-1565 (Fed. Cir. October 20, 2008) (slip op. at 3). For at least the foregoing reasons, *Schwier* fails to teach or suggest at least one element of independent Claim 1. Therefore, *Schwier* does not anticipate Claim 1 under 35 U.S.C. § 102. Reconsideration is respectfully requested.

DEPENDENT CLAIMS 2–29

Each of Claims 2–29 depends from Claim 1, and includes each of the above-quoted features of its parent claim by dependency. Thus, *Schwier* also fails to teach or suggest at least one feature found in Claims 2–29. Therefore, *Schwier* does not anticipate Claims 2–29. Reconsideration of the rejection is respectfully requested.

In addition, each of Claims 2–29 recites at least one feature that independently renders it patentable. For example, **Claim 9** recites, among other elements, steps of “generating, based on the original format, a set of conversion instructions to convert the second document into said second merge document;” and “passing the set of conversion instructions from the merge utility to the first document authoring application.” The Office Action alleges that such steps are taught in *Schwier* at FIG. 2 and column 4, lines 15–20. The Office Action is mistaken, for at least the reason that, while *Schwier* may teach that an application may send conversion instructions to a PCL convertor 18, PCL convertor 18 is not a “first document authoring application.” Moreover, the application does not perform this step until after it has merged variable and static data, whereas Claim 9 recites that this step occurs as part of converting the second document prior to merging the first merge document and the second merge document.

To expedite prosecution in light of the fundamental differences already identified, further arguments for each independently patentable feature of Claims 2–29 are not provided at this time. Applicants reserve the right to further point out the differences between the cited art and the novel features recited in the dependent claims.

III. CONCLUSION

For the reasons set forth above, all of the pending claims are now in condition for allowance. The Examiner is respectfully requested to contact the undersigned by telephone relating to any issue that would advance examination of the present application.

A petition for extension of time, to the extent necessary to make this reply timely filed, is hereby made. If any applicable fee is missing or insufficient, throughout the pendency of this application, the Commissioner is hereby authorized to charge any applicable fees and to credit any overpayments to our Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Date: March 19, 2009

/KarlTRees#58983/

Karl T. Rees, Reg. No. 58,983

2055 Gateway Place, Suite 550
San Jose, CA 95110
(408) 414-1233
Facsimile: (408) 414-1076